Aberystwyth University

An artificial immune system for continuous analysis of time-varying data

Neal, Mark

An artificial immune system for continuous analysis
of time-varying data

Dr. Mark Neal,
Department of Computer Science,
University of Wales,
Aberystwyth,
Ceredigion, SY23 3AF
U.K.
Email: mjn@aber.ac.uk

Abstract

This paper presents an artificial immune system (AIS) which produces artificial immune networks that are meaningful, of a bounded size and dynamic over a very large number of data presentations. This behaviour had proved elusive up to this time but has now permitted the application of the AIS to situations requiring continuous learning. It also removes the need to decide when to stop training an AIS. The new version of the algorithm is described, and results are presented for analysis of static and dynamic versions of a trivial two-dimensional data set and Fisher's Iris data. It is argued that the changes made from previous versions of the "resource limited" algorithm are in keeping with the goals of remaining true to the immune system analogy and making the system as simple as possible.

1 INTRODUCTION

The human immune system is a complex natural defence mechanism that recognizes and responds to the presence of foreign substances (pathogens). The response elicited depends on the previous experience of the immune system in question. Invaders that display antigens (features of pathogens) that have been experienced previously elicit a more rapid and more powerful response. This flexibility enables the immune system to remove a huge variety of infections, many of them novel to the immune system in question. This ability to learn and respond to a wide variety of similar but different pathogens has roused the interest of Artificial Intelligence researchers who wish to learn from, emulate and exploit *artificial* immune systems.

There are several competing theories as to how the human immune system achieves the adaptability and flexibility that allows it to function so effectively. The existence and participation of the bone marrow, B-cells and T-cells in the process is beyond dispute. The ways in which these entities reproduce, clone and mutate is still a fertile field of study for immunologists. Computer scientists have for many years used evolutionary computing as a stock in trade (see Goldberg 1989), and thus understand something of how to deal with simulations of simple versions of these types of activity. The added interest of the immune system is in the mechanism that makes it so effective and so rapid in adapting, more rapid than organism level evolutionary adaptation.

Of the various mechanisms suggested, the network theory (see Jerne 1974, Perelson 1989), still very contentious in immunology circles, stands out as a tractable and familiar way to try to improve upon the performance of the standard genetic algorithm. AI has often resorted to networks of one type or another as mechanisms that can be made to exhibit emergent behaviour in a reliable, comprehensible and visually presentable way. Thus we have been working with models of immune systems based on network structures with B-cells as the primary unit (see Timmis et al. 1999, Timmis et al. 2000 and Timmis et al. 2001).

2 REAL AND ARTIFICIAL IMMUNE SYSTEMS

At this point a brief summary of some of the relevant terms and how they apply to real and artificial immune systems is appropriate:

i) Pathogen: for the biological immune system a pathogen is usually a foreign body such as a virus, bactaerium, fungus or other parasite. For an artificial immune system a complete data item represents a pathogen.

ii) Antigen: a real antigen is a substance which elicits a response from lymphocytes. These are often toxins or proteins which are characteristic of particular types of pathogen. In the artificial immune system a field within a data item with a particular value is comparable; as it is particular values in particular fields which stimulate the nodes in an artificial immune system.

iii) Lymphocytes: are the white blood cells in the real immune system which are responsible for the destruction of pathogens. B-cells and T-cells are two types of lymphocyte. In our artificial immune system B-cells are not represented individually,

but gathered together using the concept of the artificial recognition ball (ARB) as is described below (see section 2.4).

iv) Innate versus adaptive immunity: innate immunity does not change throughout the lifetime of the individual and relies on different mechanisms from adaptive immunity which is what we are concerned with and wish to emulate in our artificial immune systems.

2.1 INITIAL INNOCULATION

The adaptive human immune system is primed at a very early stage in various ways including from the mother's milk and via vaccinations. For the human these very early additions to the immunological repertoire often mean the difference between life and death. Clearly the ability to bootstrap the immune system before any dangerous pathogens are missed is an essential feature of any immune system. Fortunately the effects of failure in AI systems tend to be less drastic than in the human body, but nonetheless the sensitivity of any immune system, real or artificial, to its initial pre-programmed repertoire is of the utmost importance. If it is necessary to pre-program with a very large number of antigens, and the system is not capable of dealing with antigens significantly different from those in the initial innoculation then this is not satisfactory. In fact the less that is necessary to begin with, the better.

2.2 PRIMARY RESPONSE

The primary response of an immune system is provoked when an antigen not previously encountered is detected. The bone marrow will generate a large number of B-cells, in the expectation that some of them will be able to deal with the infection, and will thus take over the production of more and more effective antibodies. After the response has cleared the infection, some of the more effective B-cells produced will remain in the body ready to respond the next time a similar infection occurs.

This part of the process is recognized as a learning phase in which previously unseen patterns are stored for later recall. The way in which the B-cells that remain in the system are maintained, and do not die off is of fundamental importance and is where the network theory provides one of several possible answers.

2.3 SECONDARY RESPONSE

The secondary response is the response elicited when a familiar antigen is detected. Those B-cells already present in the body which are well adapted to dealing with the antigen will reproduce very rapidly to deal with the infection.

The secondary response can be seen as the recall phase in the artificial immune networks presented.

2.4 THE IMMUNE NETWORK THEORY

The immune network theory proposes that the B-cells in the body interact with each other to maintain the immune memory. The mechanism proposed is that B-cells which are capable of recognising similar (but not necessarily identical) pathogens are also capable of recognising and stimulating each other (see Farmer et al. 1986). Thus a dynamic feedback mechanism can maintain parts of the immunological memory which are not frequently stimulated. Clearly however not all B-cells have sufficient stimulation to survive indefinitely and thus some will die out.

In the human immune system T-cells both perform a surveillance role and interact with B-cells which complicates the mechanism somewhat. In our artificial immune system the role of T-cells is currently ignored.

In the real immune system there are very large numbers of identical B-cells to deal with each type of infection. In an artificial system such repetition can be coded without representing all the identical cells individually. Fortunately the concept of a *recognition ball* which represents a region of antigen space that is covered by a particular type of B-cell can replace the repetition of individuals (Perelson 1989).

So our AIS consists of a network of *artificial recognition balls* which are linked together if they are close to each other in antigen space. Pathogens (data items) can be considered to be points in this antigen space, and thus proximity can be defined as a simple distance function. When a data item is presented to the network the node which is the most stimulated produces clones of itself, some of which are mutated to increase the diversity of the network's recognition capabilities. The stimulation level of each node is calculated based upon its reaction both to the data items and to those nodes to which it is connected (see section 4.1). Thus nodes which are severely mutated into remote regions of the antigen space (and thus sparsely or totally disconnected) will not survive unless they match data items which are not presently covered by the network in which case they will expand its repertoire.

3 BACKGROUND

In a previous publication (see Timmis et al. 2001) we presented a resource-limited version of the AIS as a step toward a continuous learning version of the AIS presented in (see Timmis et al. 1999, Timmis et al. 2000). This previous work was motivated by the need for an AIS that did not rely on the arbitrary selection of the number of times that a data set should be presented to it, and the realisation that any AIS that did require such control was not a good model of a biological immune system. There were however several problems with the solution that we proposed:

i) the mechanisms which governed the resource allocation were centralised in a very artificial way,

which was contrary to the distributed nature of the original AIS

ii) there was no "inertia" effect bound to the resources. Thus an ARB could gain or lose all of its resources in one pass through the network, which is quite unlike the biological immune system which takes time to build up immunity and time to lose it again.

iii) The nature of the calculations performing the resource allocation required the normalisation of the stimulation levels, which lead to some inelegant, lengthy and unnecessarily complex calculations after every iteration

iv) After several passes through the data set in question the network would begin to degenerate and fail to represent some of the data items

v) The algorithm did not lend itself to a genuinely continuous mode of operation as resource allocation was performed after each pass through the data set. This required an epoch-based (synchronous update) approach which creates a variety of problems if the network is to be used in a continuous mode.

After several attempts to modify the resource allocation mechanism it became clear that these problems were quite severe and were leading to a complex and arbitrary set of solutions. Thus a different approach was taken based on a simpler mechanism used after every data item presented.

4   THE SSAIS

This new approach lead to the self-stabilising artificial immune system (SSAIS) presented here. Artificial recognition balls (ARBs) are still used as the basic component of the network, and they are still linked together in the same way. The network affinity threshold is also calculated in the same way and serves the same purpose as in the original systems. The SSAIS differs from the resource limited artificial immune system (RLAIS) in several ways. The most important difference is that there is no fixed quantity of resources to be distributed centrally between the ARBs. The concept of resources is still present, but in an altered form. In the RLAIS the resources were allocated to ARBs by order of and in proportion to stimulation level. In the SSAIS resources are dealt with locally by each ARB. An ARB increases its own resource allocation each time it registers the highest stimulation for an incoming data item. The ARB increments its resource holding by adding its current stimulation level. Additionally, each time a data item is presented the resource level of every ARB decays geometrically. The balance between the decay of the resource level and the occasional boost received when an ARB "wins" is quite robust, and results in more densely populated areas of the data space supporting larger numbers of ARBs and more sparsely populated regions fewer ARBs. This results in emergent behaviour that is very similar to that of the original AIS and the RLAIS, but without the "one shot"

constraint of the former and the normalisation, synchronous update and sorting requirements of the latter.

4.1   THE STIMULATION FUNCTION

In order to bound the growth of the resource level in any ARB (and thus in the network as a whole) it was necessary to bound the stimulation level. The simplest way to achieve this is to make a small modification to the ARB stimulation function. The stimulation function in previous systems (see Timmis et al. 2000) was made up of three components:

i) An excitation factor, ps based linearly on the Euclidean distance to the current data item (p):

$$ps = 1 - dis(p)$$

ii) An excitation factor, ns based on the distance to the neighbours around the ARB:

$$ns = \sum_{x=0}^{n} 1 - dis(x)$$

iii) A suppression factor, nn based on the distance to the neighbours around the ARB:

$$nn = -\sum_{x=0}^{n} dis(x)$$

In all equations the function $dis(a)$ returns the Euclidean distance between the current node and the item a; and n represents the number of neighbours at the current node.

These components are simply summed. The second and third components are based on the neighbours of the ARB, and there is no limit to the number of neighbours an ARB can have. This poses a problem in the form of the potential for unbounded growth. Two variants on this stimulation function were experimented with. The first of which is the most obvious and is simply the same as above, but with parts ii) and iii) divided by the number of neighbours. This succeeded in bounding the growth of the resource levels in the network, but resulted in networks which had one extremely dense and active region and other totally static sections which were much less dense remainders of the original network created from the initialisation data. In order to examine this behaviour a second simpler function was used with surprisingly effective results. The neighbour suppression factor was discarded completely and only the excitation retained. This resulted in a simpler stimulation function made up of only two parts which are summed:

i) An excitation factor, ps based linearly on the distance to the current data item:

$$ps = 1 - dis(p)$$

ii) A normalised excitation, ns factor based on the distance to the neighbours around the ARB:

$$ns = 1/n \times \sum_{x=0}^{n} 1 - dis(x)$$

When used within the scheme presented here, this function yielded networks which attain a "dynamic stability" with all parts of the network producing some clones (see below), and varying their topology a little at a time, whilst retaining the overall structure and distribution throughout the data space.

4.2   ALLOCATING RESOURCES

In this version of the immune network algorithm, *resources* are simply recorded as a numerical value associated with each node. This number is used both to decide when to remove a node from the network (when the resources fall below a minimum threshold) and to decide how many clones to produce (more resources implies more clones). Whilst there is no longer a central notion of resource availability, it is still appropriate to think of the ARBs being limited by available resources. In this system the ARBs allocate their own resources only when justified by reacting the most strongly to a data item. The level of resources at an ARB that is not the most stimulated by data item $(i+1)$ is geometrically decaying with each data presentation, thus:

$$R_{(a)(i+1)} = dr \times R_{(a)(i)}$$

where $R_{(a)(i)}$ represents the level of resources present at ARB a after the presentation of i data items and dr represents the rate at which the resource level at an ARB decays. The level of resources at the ARB which is the most stimulated by data item $(i+1)$ will be:

$$R_{(a)(i+1)} = dr \times (R_{(a)(i)} + SL_{(a)(i+1)})$$

where $SL_{(a)(i+1)}$ represents the stimulation level of ARB a (as defined in section 4.1) after the presentation of data item $(i+1)$. Thus when an ARB is the most stimulated for an incoming data item it gives itself a boost in its resource level. These two conflicting effects balance to ensure the survival of ARBs that regularly have the highest stimulation level and the gradual demise of those that do not. The decay rate scalar dr provides an easy control over the size of the networks produced. The values used for dr in this work were 0.999 for the trivial data set and 0.9995 for the Iris data. Some initial experimentation with these values was undertaken which seemed to indicate that the value of dr is a sensitive control for the size of the population.

4.3   POPULATION CULLING

After each data item is presented to the network any ARBs that have resources less than a fixed threshold value (the *mortality* threshold) are removed from the population. The threshold value used in this work was 0.6 for all networks regardless of the data set in use. This was an arbitrary choice, and further work is required to ascertain the sensitivity and range of values for this parameter. The networks produced do not seem to be particularly sensitive to the threshold at which nodes are culled. The values for *mortality* and the multiplier for the resource level for new clones are also arbitrary and require further investigation.

4.4   CLONING MECHANISM

The cloning mechanism for the SSAIS is slightly different from previous systems. When an ARB is the most active it is allowed to undergo cloning. The ARB produces clones at a rate which is proportional to the resource level at the ARB. The number of nodes produced is calculated as follows:

$$nc = R_{(a)(i)}/(mortality \times 10)$$

where *mortality* is the minimum resource level that a node can have before being culled. This is because each clone that is produced is assigned *mortality* $\times$ 10 resources from the ARB's pool of resources. As each clone is produced its data fields are mutated with a fixed probability (the mutation rate). The mutation rate was fixed throghuout this work at 0.1%. If the clone is mutated then it gives rise to a new ARB with *mortality* $\times$ 10 resources. If it is not mutated then the resources are returned to the parent ARB. The new clones are incorporated into the network and the processing of the data items continues.

4.5   THE ALGORITHM

Prior to the commencement of training the network an initial innoculation of ARBs must be provided. For the work presented here 10 ARBs were used to initialize the network for the trivial data set, and 30 ARBs were used for the iris data set. These numbers were used because they represent 20% of the number of items in each data set. The items from the data sets were simply every fifth one in whatever order they happened to be. Initial experimentation with different initial innoculations indicated no significant difference in behaviour when using different sub-sets of either data set.

Thus bringing all the above elements together, we can summarise the continuous algorithm as follows:

i)   Innoculate the network with a random set of ARBs

ii)   present a data item to all the nodes

iii)   find the node with the highest activation

iv)   allow this node to increase its resource level

v)   deplete resources at all nodes

vi)   cull nodes with less than threshold resource level

vii)   allow highest activation node to clone

viii)   relink the network with new clones

ix)   return to ii)

5   EXPERIMENTS AND RESULTS

Results for two data sets in two different modes are presented. The first set of data consists of 50 two dimensional data items arranged in two clusters (see figure 8a). This was designed as a development tool to allow simple visualisation of ARB positioning in a well understood data set. The second set of data is Fisher's famous Iris data (see Fisher 1936) which provides a well

known benchmark data set with understood properties and some more challenging characteristics. The data consists of 150 four dimensional data items belonging to three categories, each of which represents a variety of Iris. A principal component plot (see Everitt 1974) of the first two principal components is presented in figure 8b. Both data sets were presented to the AIS as continuous streams of data which wrapped around each time the end of the data set was reached. The first two experiments were carried out using 20% of the data items as an initial innoculation and thereafter presenting all the data items from the outset. This type of analysis will be referred to from here on as *complete*. The last two experiments took one of the clusters from each data set and used 20% of this reduced set as an initial population and then trained for 250,000 data item presentations to demonstrate initial stability. Then the remainder of the data set was introduced and the network trained for a further 750,000 presentations to demonstrate the new stable state with the increased repertoire. This type of analysis will be referred to from here on as *incremental*.

5.1 COMPLETE ANALYSES

The complete analyses were carried out over 1,000,000 data item presentations to demonstrate long-term stability. The networks settle to a quasi-steady-state much more rapidly.

5.1.1 Trivial data

The networks produced for the complete analysis of the trivial data set very rapidly settled down to two distinct clusters of ARBs with the occasional appearance and disappearance of small outlying clusters or singlets which were rapidly culled (see figure 1). The network was examined at a large number of points during training and seemed to vary very little, although the addition and culling of clones occurred throughout (see figure 2).
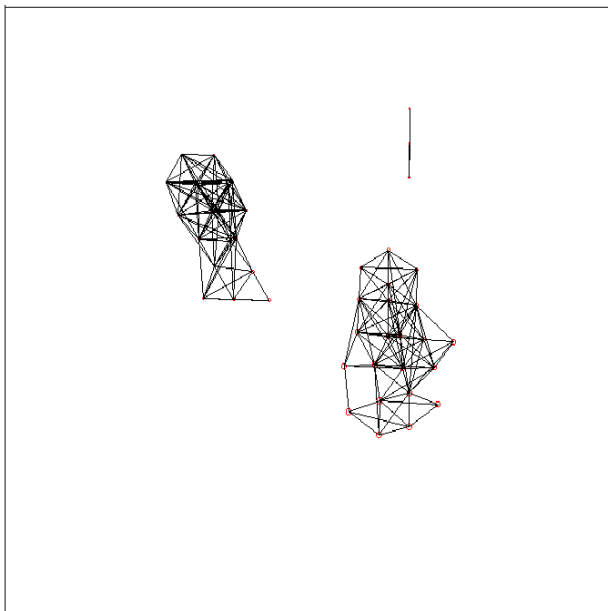

Figure 1: The network produced for the trivial data set after 30,000 data items have been presented

The size of the network settled down to between 40 and 55 quite rapidly. Variations in size and structure continued but did not vary the basic structure of the network after approximately 1000 data items had been presented and processed. Slight variations in size and structure are due to the stochastic nature of the network introduced by the cloning and mutation mechanism.
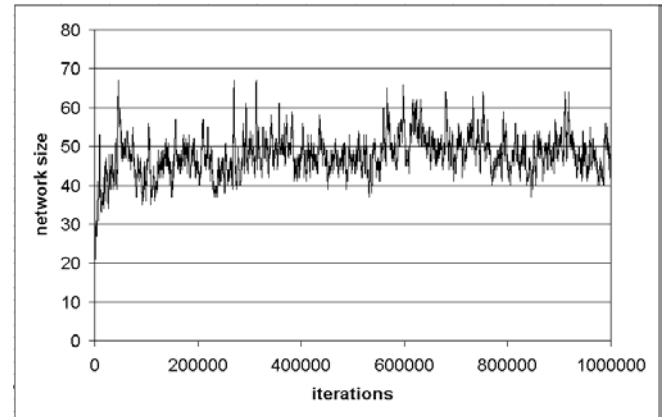

Figure 2: Size evolution of the network running on the trivial data set

The input space was densely populated in regions containing high densities of data throughout training. Regions of lower density outside the clusters of data were either devoid of ARBs, or supported small clusters of 1,2 or 3 ARBs for brief periods. These appeared due to the mutation of clones from the two groups.

5.1.2 Fisher's Iris data

This data set provides an interesting test for any data analysis technique as it consists of one clearly separable class of data (the Setosa class), and two slightly intermingled classes (the Virginica and Versicolor classes). A conventional Principal Component Analysis plot of the data shows this quite clearly in figure 8b. The network produced by the SSAIS after 350,000 data item presentations is shown in Fig. 3.

The evolution of this network was allowed to run on for 1,000,000 data presentations in order to examine the long-term behaviour of the network. The shape of the network was examined at various points and after about 100,000 iterations there were no major alterations in structure with a separate group for the Setosa class and an elongated group for the other two classes.
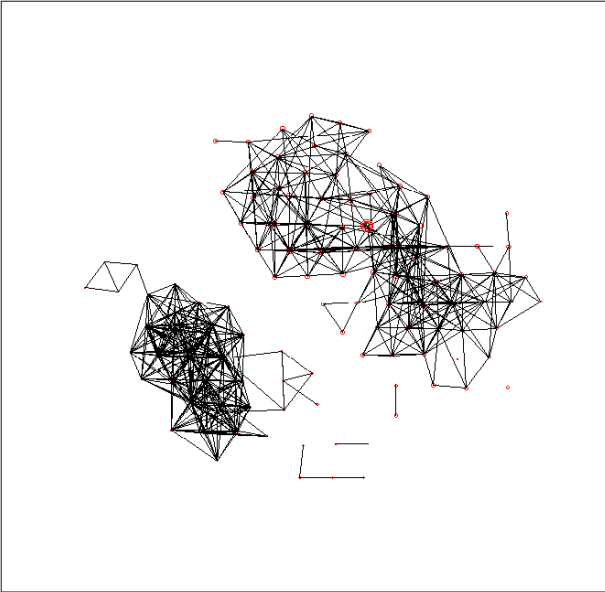
Figure 3: Network produced for the Iris data after 350,000 data item presentations
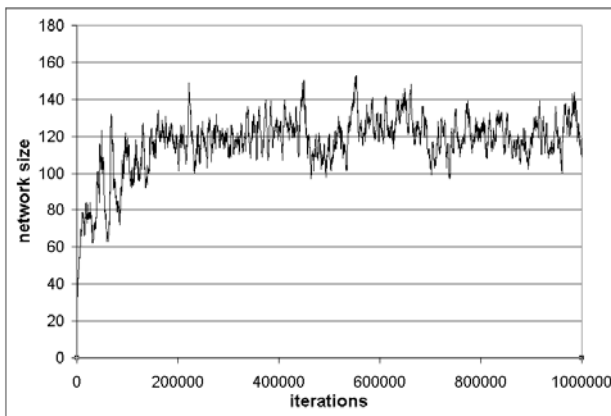

Figure 4: Size evolution of the network running on Fisher's Iris data set.

 The long-term evolution of the size of this network is shown in Figure 4. The trace shows very rapid growth initially followed by gradual growth until about 150,000 iterations. Thereafter the network has a relatively stable size that varies by about 20 nodes either side of 120. This steady but dynamic behaviour is desirable as it indicates continuing introduction and maintenance of diversity within the network, whilst retaining reasonable coverage of the data space over a very long period. The enduring shape of the network can be seen in figure 5 which shows the final state of the network after 1,000,000 iterations.
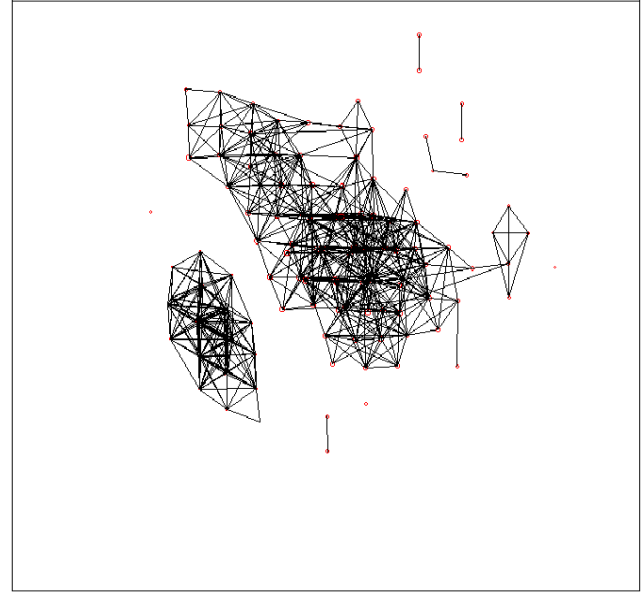
Figure 5: Network produced for Iris data after 1,000,000 data items have been presented.

Thus the networks produced throughout training on the Iris data cover the data space well, and reflect the nature of the groupings in the data.

5.2   INCREMENTAL ANALYSES

The incremental analyses were carried out over 1,000,000 data presentations in order to demonstrate the stability of the networks in their new configurations. Typically the behaviour of the networks settles down much more rapidly than this.

5.2.1   Trivial data

For the incremental analysis of the trivial data set the network was initialized with 5 of the 25 data items from the cluster close to the origin (see figure 8). The network was then trained for 250,000 data presentations with the members of only that cluster. The data being presented was then expanded to include the second group of data which is centred around the point (0.8,0.8). The size evolution of the network is shown in figure 6.

The network size can be seen to stabilise at the beginning of training at a size of between 30 and 45 nodes whilst only the first cluster of data is being used. The second cluster of data is introduced after 250,000 iterations after which the network takes about 200,000 more iterations to begin to cover the new data cluster. Examination of the intermediate networks produced shows little development of the network into regions which cover the new data. This seems to be due to the relatively confined region which the network covers before the second cluster is introduced. This lack of diversity in the network makes it unlikely that any mutated clone with only a single mutated antigen will be close enough to the new data items to survive. Thus the chance generation of several clones into the same region is required in order for the colonisation of the newly populated region

of input space to begin. Once a start has been made, the new region is rapidly covered quite effectively. This is shown by the increase in population size at 500,000 iterations. See figure 9 for network evolution.
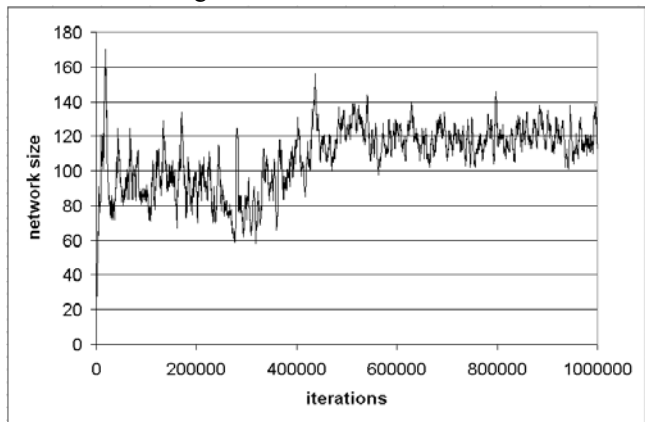

Figure 6: Size evolution of the network running on the trivial data set with introduction of second cluster at 250,000 iterations.

5.2.2 Fisher's Iris data

For the incremental analysis of Fisher's Iris data the network was initialized with 10 of the 50 Setosa class (see figure 8). The network was then trained for 250,000 data presentations with members of only that cluster. The data being presented was then expanded to include the other classes of data (Virginicas and Versicolors) which form a clearly distinct cluster. The size evolution of the network is shown in figure 7.
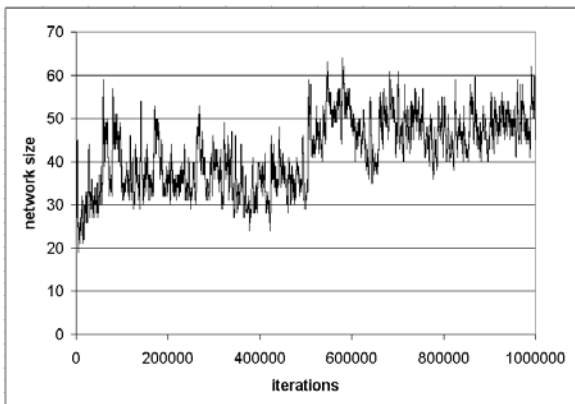
Figure 7: Evolution of network size for incremental analysis of Fisher's Iris data.

The network can be seen to have settled to a reasonably constant size of between about 70 and 110 when training on only the Setosa cluster (before 250,000 iterations). Subsequent to the introduction of the second cluster of data the network undergoes some fairly rapid changes. Initially there is a short period (between 250,000 and 300,000 iterations) of decline in size of the network. Then there is a period of quite rapid growth until about 450,000 iterations after which the network settles down to a fairly steady size of between about 105 and 130 nodes. Prior to the

introduction of the second group of data the network consists of a single highly connected cluster of nodes. Upon the introduction of the additional data the network spreads out into a more complex structure before several chunks split off from the initial cluster and reform into a second large highly connected cluster. The ultimate shape which the network assumes is very similar to that produced by the complete analysis presented in section 5.1.2 (see figure 5). Snapshots of the network evolution throughout the incremental analysis are shown in figure 10.

6   DISCUSSION

The goal of this work was to create a genuinely stable, adaptive and continuous AIS. The changes that were introduced grew out of the realization that the shortcomings of the RLAIS (Timmis et al. 2001) stem from two fundamental problems: the nature of the resource allocation mechanism and the explicitly non-continuous nature of the epoch based update mechanism. The latter problem of assuming that there was an obvious point at which to stop presenting data items and perform an "update" was very simple to deal with. This just involved re-examining the algorithm and making sure that every operation could be carried out after the presentation of every data item. Most of the components of the system lent themselves readily to this approach, and as the resource allocation scheme was under scrutiny, problems with that aspect and the closely related problem of when and how much to clone were redesigned to fit the new regime. Successfully altering the resource allocation scheme required a little more thought. The fields of genetic algorithms and artificial life have taught many lessons about the nature of emergent behaviour in such systems, one of the most basic being that decentralization of control mechanisms usually leads to more interesting behaviour (see Johnson 2001). This led to the (now obvious) idea of devolving resource allocation to the ARBs, and adjusting the stimulation function to facilitate this. Thus now the only centralized function is that of choosing the winning ARB from the network. Finding the winner locally in the network would probably be possible, but unnecessarily complex and somewhat pedantic, especially as it could be argued that the bone marrow is a centralized controller of some importance in the biological immune system. Other mechanisms which allocate resources based on "local" winners were briefly examined and may be the subject of further research.

The time lag between the introduction of a new region of input data and the network covering the new region of the data space is disappointing. This is especially evident in the incremental analysis of the trivial data set. It seems clear that this lag is primarily due to a lack of diversity in the network. The network is slow to regain the diversity required to cover the new region due to the mutation and cloning mechanism, which is likely to produce mutations with only one data field different from the parent ARB. Thus it seems that examining more effective cloning and mutation mechanisms for the primary response would be of great interest. These are likely to involve an *artificial bone*

marrow that produces random antibodies when a poorly recognized pathogen is detected.

Control of the size of the network is to some degree removed from the domain of the user of the SSAIS, but clearly not entirely. The number of ARBs with which the network is initialized provides an initial point from which the system can evolve and thus provides a short-term control although the *mortality* constant and *decay rate* are far more sensitive and control the long-term meta-dynamics of the networks. The *mortality* constant provides a very coarse control which is unlikely to be changed in practice. The *decay rate* however provides a much finer control over the size of networks produced. Precisely how the size of the network relates to the *decay rate* will vary depending on at least the density of the data points in the input space, and the frequency of repetition of similar items. With fixed data sets the latter of these is simply the number of items in the set. The former is hard to measure, and its effect harder still. Some type of automatic and dynamic control of the *decay rate* would be extremely useful and remove a potential fudge factor.

7   FUTURE WORK

A number of pieces of work will flow directly from this approach to the construction of artificial immune networks:

i)   The testing of the algorithm on some more complex data sets from the real world. This will enable some detailed comparisons with other techniques to be made, as well as to verify that the behaviour seen with the data sets presented here is repeatable.

ii)   Running the algorithm on a continuously varying data source rather than fixed data sets presented many times to examine the flexibility of the representations formed and the rate at which the networks can track varying input.

iii)   Creating an efficient and well engineered implementation of the algorithm. This will offer some performance increases, although performance has not proved to be a problem, as well as providing a stable software platform on which to base further experiments.

iv)   Examining more realistic and intelligent cloning and mutation mechanisms. There is evidence that biological immune systems employ some very well controlled and directed cloning and mutation mechanisms, none of which are exploited here (see Kepler et al. 1993). Significantly different and potentially more useful behaviour could be expected if some methods such as these were applied.

8   CONCLUSIONS

The algorithm presented here generates networks of a bounded size over an indefinite number of data presentations and updates. The networks produced are continually changing whilst retaining good coverage of the input space and some diversity via the mutation mechanism employed. No central control in the form of a resource allocator is required which holds true to the distributed nature of the networks under construction. The system also has the advantage of being conceptually simpler than the previous resource limited artificial immune system. The dynamic stability displayed is a better model of the immune system than previous work presented and shows great promise for applications requiring analysis of continuously changing data sets with minimal intervention in the learning process.

References

B. Everitt (1974). Cluster Analysis, Heinemann, London.

J.D. Farmer, N.H. Packard (1986). The immune system, adaptation and machine learning, *Physica 22D*, 187-204.

R.A. Fisher (1936). The use of multiple measurements in taxonomic problems, *Annual Eugenics, Part II 7,* 179-188.

D.E. Goldberg (1989). Genetic algorithms in search optimization and machine learning, Addison Wesley.

N.K. Jerne (1974). Towards a network theory of the immune system, *Annals of Immunology 125C,* 373-389.

S. Johnson (2001). Emergence, the connected lives of ants, brains, cities and software, Penguin Press, London.

T.B. Kepler, A.S. Perelson (1993). Somatic hypermutation in B cells: an optimal control treatment. Journal of Theoretical Biology, 164, 37-64.

A. Perelson (1989). Immune network theory, *Immunological Review 110*, 5-36.

J. Timmis, M.Neal, J.Hunt (2000). An artificial immune system for data analysis, *Biosystems 55 (1/3),* Elsevier, 143-150.

J. Timmis, M.Neal, J.Hunt (1999). Data analysis with artificial immune systems, cluster analysis and kohonen networks: some comparisons. In *Proceedings of the International Conference on Systems Man and Cybernetics,* IEEE, Tokyo, Japan,. 922-927.

J. Timmis, M. Neal (2001). A resource limited artificial immune system for data analysis, *Knowledge-Based Systems 14,* Elsevier, 121-130.
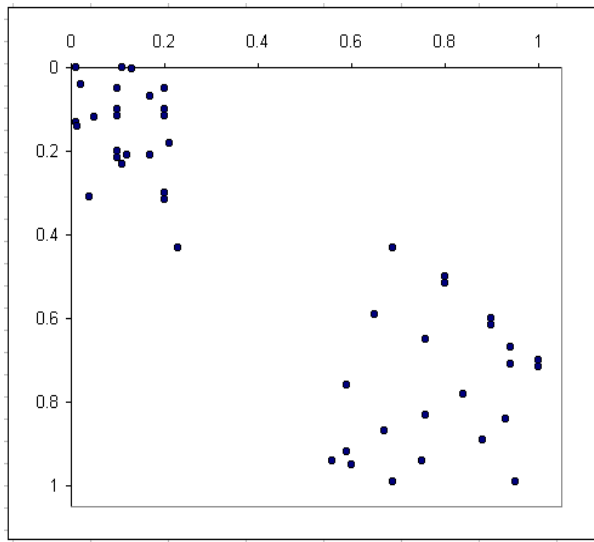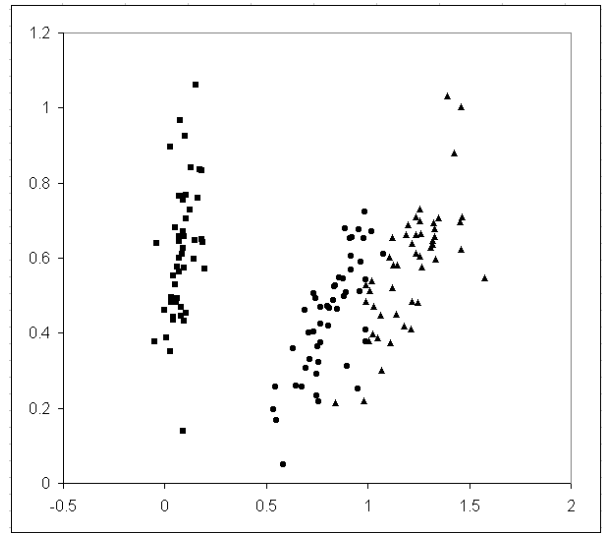
Figure 8: a) Two-dimensional trivial data set

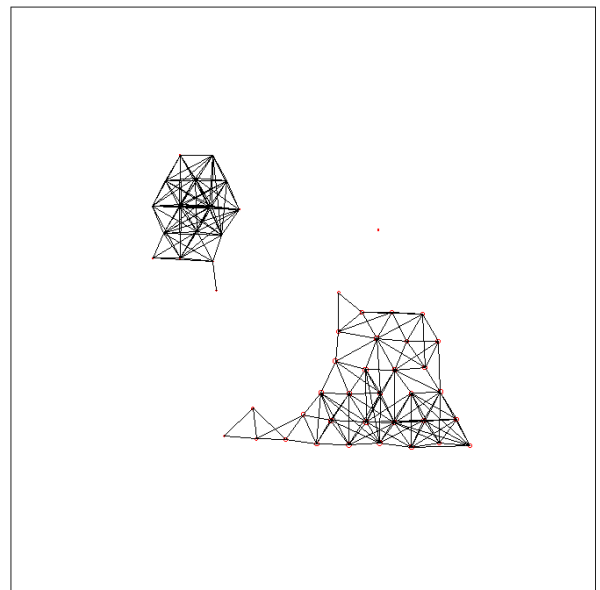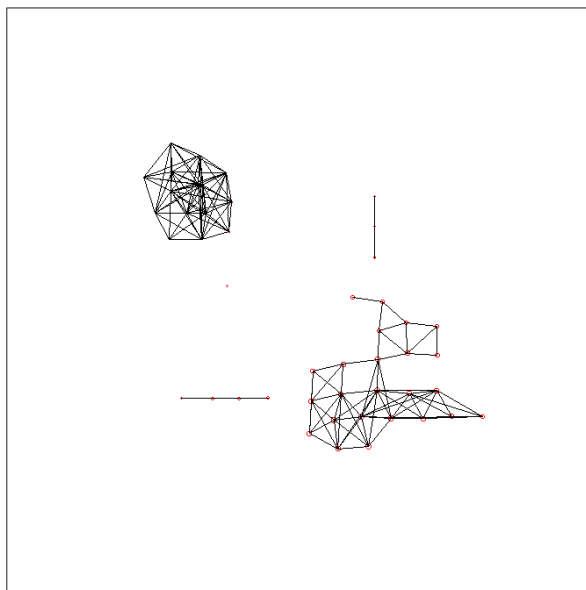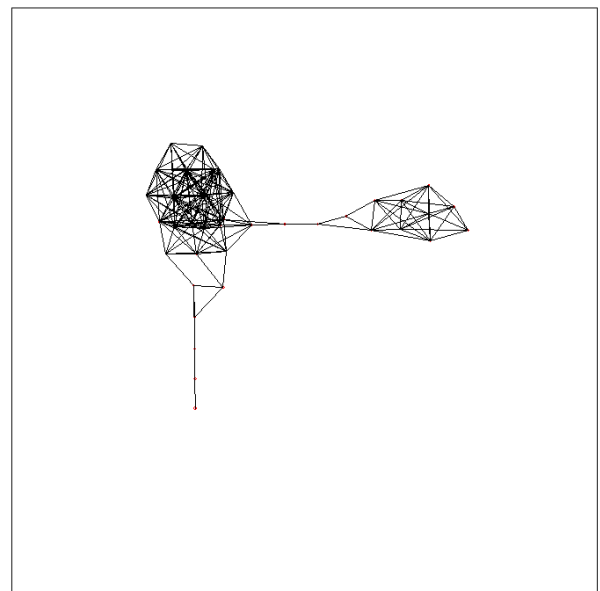b) Principal component plot of Fisher's Iris data. Setosa square, Virginica round, Versicolor triangular.
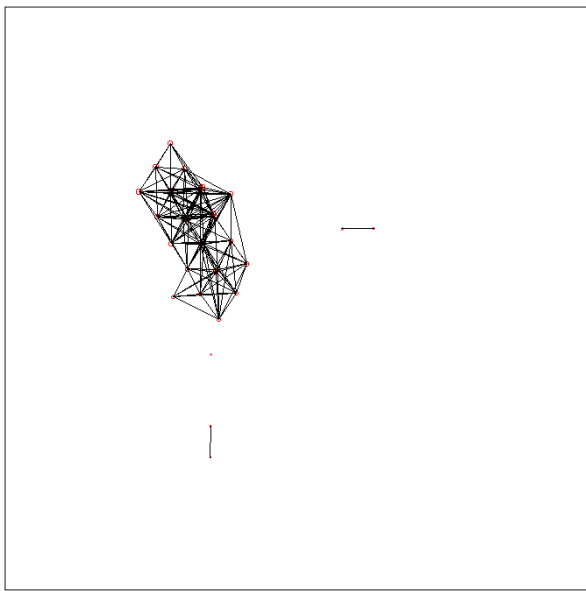

Figure 9: Network evolution during incremental learning of trivial data set. Series evolves top left to bottom right.

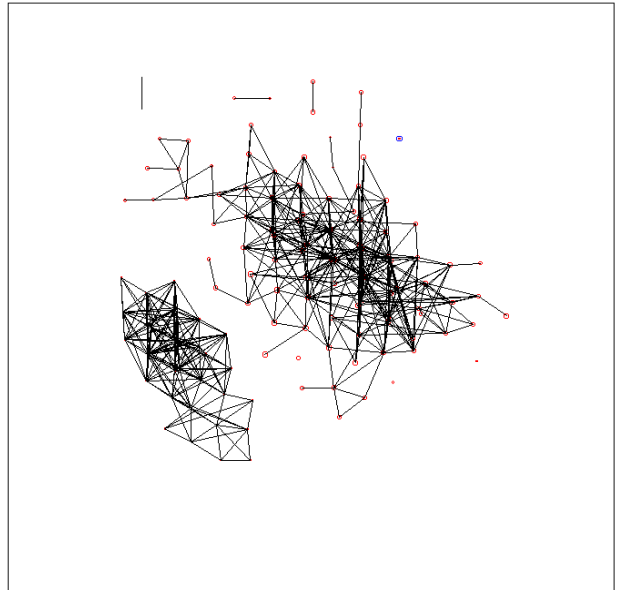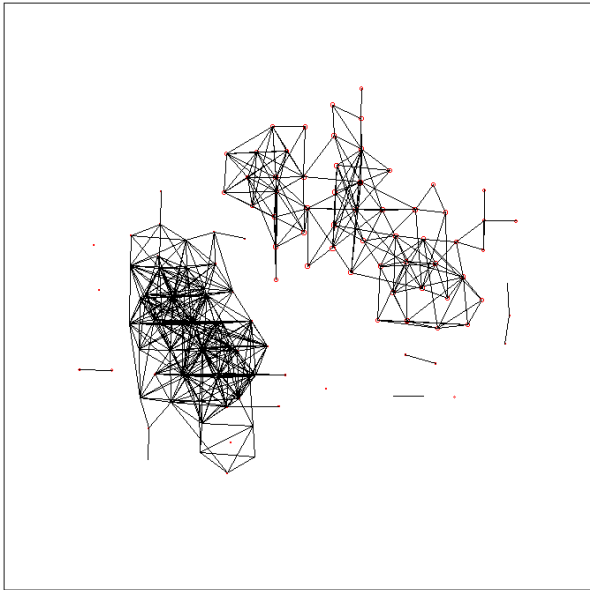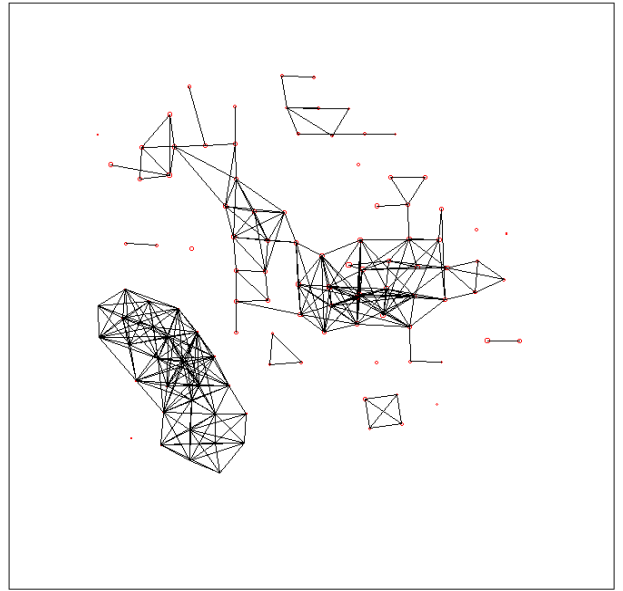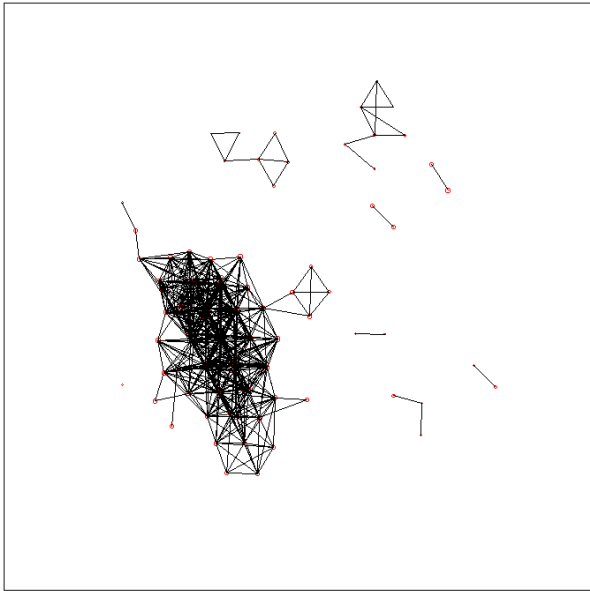Shots taken at 250,000,300,000, 400,000 and 450,000

Figure 10: Network evolution during incremental learning of Fisher's Iris data. Series evolves top left to bottom right.

Shots taken at 500,000, 550,000, 600,000 and 700,000